3GPP TSG SA WG3 Security — S3#32 **S3-040121**
09 - 13 February 2004
Edinburgh, Scotland, UK

Agenda Item: 6.9.2 GBA

Source: ~~Nokia~~Siemens

Title: Remarks on S3-040042 by Nokia and replies by Guenther Horn (Siemens), dated 6 Feb 2004

Document for: Discussion/Decision

Guenther: Please find my replies to the comments inline.

This is a response contribution to previously submitted Tdoc S3-040042:" Multiple key derivation in a Generic Bootstrapping Architecture - Pseudo-CR", by Siemens.

Guenther: it may be helpful to read S3-040042 in conjunction with S3-040041 and S3-040044.

The previous discussion of the key derivation

The basic idea in S3-040042 was seen in previous meeting. The discussion raised two open issues:
Guenther: the whole question you address in this section has nothing to do with the fact whether multiple keys Ks_NAF (as proposed in S3-040042) are derived from one Ks or not, as I try to show in the following. Please remember that key derivation as specified in TS 33.220 v100, addresses the problem of separation of application domains.

(1) The key-synchronization problem: The key lifetime for different NAFs should not be mandated equal due to the individual nature of each service. Guenther: I agree. For instance, currently e-commerce often uses one-time-password for one session, but for some other services there is never need to change the access secret or password. It is not good principle that lifetime of the key must be equal to all NAFs.
Guenther: they do not have to be equal. In fact, the lifetimes of the keys Ks_NAF shared between UE and NAF can be different. This is how it works according to the current version of TS 33.220 and S3-040041: when the NAF indicates to the UE that a key change is required then the UE initiates a run of the Ub-protocol (http digest aka) to obtain a new key Ks. Then the key Ks_NAF is derived for the requesting NAF. But cf. the proposed new text from S3-040041 to clarify the current version of TS 33.220: "When a new Ks is obtained over the Ub interface and a key Ks_NAF, derived from one NAF_Id_n, is updated, the other keys Ks_NAF, derived from different values NAF_Id_n, stored on the UE shall not be affected." S3-040041 does not assume the derivation of multiple keys Ks_NAF from one Ks. By introducing multiple key derivation, as described in S3-040042, the property of having different lifetimes for different keys Ks_NAF is retained. The only change introduced by multiple key derivation in S3-040042 is that, for a NAF with which the UE does not yet share a key, no new Ks needs to be generated. Rather, the Ks_NAF can be derived from the Ks stored in the UE. The lifetime of Ks is communicated to the UE over Ub, cf. also S3-040041. The lifetime of Ks_NAF is communicated to the NAF over Zn according to Nokia's S3-040077.

(2) Different key stored in UE and NAF: how to handle, for instance, the case that UE's battery is brutely removed, or dropped, so the Ks is still valid in NAFs, but not in UE. Then UE would trigger a new round of bootstrapping procedure after re-power-on. On the other hand, the NAF who stores the old but yet still valid key would not need to communicate the BSF for retrieve another key material.

Guenther: the Ks is stored only in the BSF and in the UE, the NAF obtains only Ks_NAF from the BSF. The removal of the battery is equivalent to powering the UE down, which is addressed in S3-040041. Proposed text there: " When the UE is powered down, or when the UICC is removed, any keys Ks and Ks_NAF shall be deleted from storage." After power up, the UE has to obtain a new key Ks by running the Ub-protocol. Please remember that it was explicitly asked for in the discussion in SA3 that the keys Ks or Ks_NAF should not be semi-permanently stored in non-volatile memory for security reasons. This is implemented by the proposed new text. There is no synchronisation problem here, as, after UE power up, the UE and the BSF first establish a new Ks and a new Ks_NAF before the UE contacts the NAF. The NAF will see a new TID and then contact the BSF to fetch the new Ks_NAF.

The previous discussion of the n value as the input parameter to key derivation

Guenther: again, the question you address in this section (use of parameter n) is independent of the key derivation, whether a single key (as in TS 33.220 v100) or multiple keys Ks_NAF (as proposed in S3-040042) are derived from one Ks, as I try to show in the following. Your comments are rather related to the discussion in S3-040044.

Please note that " The purpose of the introduction of the parameter "n" is the possibility to obtain a single key Ks_NAF (and hence e.g. a single shared-key-TLS tunnel or a single http digest security session) between a UE and a NAF, even when multiple DNS names map to the same NAF." (cf. S3-040044) . If this is not considered a goal, and having multiple TLS tunnels and/or multiple http digest security sessions between the same (UE, NAF) pair is considered fine, then we do not need the parameter n.

During the Berlin meeting, Siemens contributed S3-030042, where it was proposed to define a n value as the input parameter of key derivation algorithm. During the discussion, people criticized that the fixed n value to all NAFs would bring inflexibility to the deployment of the operator's network.
Guenther: please look at my contribution S3-040044. It is explained there that the choice of n is a trade-off between flexibility in network deployment and efficiency in key management. The fact that the same n is chosen for all NAFs does not yet cause any inflexibility in network deployment.

For example, there are 2 NAFs, but 3 servers in operator1's network:
Server1.presence.operator1.com
Server2.conferencing.operator1.com
Server3.mbms.operator1.com

Suppose server 1 and 2 are configured behind a common Authentication Proxy, and they utilize the same TLS connection, the NAF_id_n for them should be the identical, thus n=2. On the other hand, for MBMS server3, a different NAF_n should be created due to residing in the different domain, so n value should be preferrable 4 (though 3 permitted also), but not 2 or 1.

Guenther: this is an example of the trade-off between flexibility in network deployment and efficiency mentioned above. You could either accept that two different keys are shared between the UE and the Authentication Proxy. (By the way this would lead to two different TLS connections only for the case of shared-key TLS, not for TLS with server certificates.) Or else, you could name your servers in a different way to reflect that some of them are behind the same proxy. E.g. you could have "server1.presence.IMSproxy.operator1.com" and "Server2.conferencing.IMSproxy.operator1.com".

Siemens argueed in SA3#31 that the n value as the input parameter of the key derivation algorithm, is associated to only ONE NAF.
Guenther: this must be a misunderstanding. See also S3-040044. By the way, it is not n, but NAF_Id_n, which is input to the key derivation algorithm.
Since the default value is n=0, i.e. FQDN of NAF will be used, the NAF and UE can generate an identical key. Therefore at least in case of default n value, the proposal works.
Guenther: if n=0, the full DNS name of the application server (AS) is used, and if different ASs sit behind the same authentication proxy (AP=NAF) , then, for each AS accessed by the user, a different key Ks_NAF is generated. In order to have the possibility to have only one key between UE and AP=NAF, n>0 must be chosen. But if having different keys Ks_NAF for one (UE, NAF)-pair is fine, then also n=0 is fine.
But the paper did not solve this issue: if the n value is not 0 for a NAF, how does the UE learn it, and how can the BSF tell the UE?
Guenther: TS 33.220 says that the BSF sends n over the Ub interface in the 200 OK message. Remember that n is the same for all NAFs.

Another alternative removes this drawback. It was contributed in SA3#31 in TD S3-030729. The UE pushes the NAF names to BSF during bootstrapping procedure, so after the successful authentication, the BSF can indicate each different n value for key generation for the intended NAFs.
Guenther: yes, this would be an alternative, but it would be a major departure from the current principles of GBA and is likely to have effects on many parts of the GBA. We have not really studied this. 729 was rejected in SA3#31. One concern is, of course, the weakness of integrity on Ub.

Below texts are exempted from SA3#31 meeting minutes (Draft_report_s3_v005):

"TD S3-030743: Key separation in a Generic Bootstrapping Architecture. This was introduced by Siemens and proposes some changes to the draft TS and a Pseudo-CR was attached to implement the proposed changes. A parameter n is proposed for use to generate keys based upon parts of the DNS name of the NAF, allowing differentiation from the full DNS name to up to the rightmost 7 parts of the DNS name. **It was clarified that AKA is always run once to derive Kc and then once again to provide the differentiable Key and that only one key is distributed to an individual NAF."**

Guenther: I acknowledge that the contributions at the Berlin meeting were based on the assumption that " AKA is always run once to derive Kc and then once again to provide the differentiable Key and that only one key is distributed to an individual NAF."

But it is certainly **permitted** to propose additional features in contributions to later meetings, and that is what Siemens is doing with contribution S3-040042. Siemens proposes in S3-030042 to allow the derivation of key Ks_NAF for different NAFs from one Ks (but only one Ks_NAF for the same NAF), for (as we believe) good reasons stated in the contribution. At the Berlin meeting, the discussion was limited to derivation of a single key, as the derivation of multiple keys on the system were not immediately clear at the time. But the specification has been progressed now and is more stable, and, provided the clarifications in the companion contribution S3-030041 (which have nothing to do with multiple key derivation) are accepted, any concerns, especially about key synchronisation, should be addressed.

Remarks to S3-040042 regarding to the two issues

Regarding to issue (2), the pseudo-CR in S3-0400042 attempts to answer with new added texts: "if no key Ks is available in the UE, the UE fetches a key Ks from the BSF over the Ub interface, it then proceeds to derive Ks_NAF."

There is no text to explain how to protect the retrieval of Ks without a new authentication procedure.
Guenther: it is meant that the UE initiates a new run of the protocol on the Ub interface (http digest aka). The choice of the word "fetch" may have been a bit unfortunate.
It's was criticized in SA3#31, regarding to TD S3-030729.

Remarks to S3-040042 regarding to n value

The S3-040042 brings up the above discussion again. When key derivation is applied to BSF, multiple Ks_NAF must be able to generated from one Ks, the only default n value does not work for all NAFs any more.
Guenther: this seems to be a misunderstanding. According to the text proposed in S3-040042, for a particular NAF1 only one key Ks_NAF1 is derived from one Ks. If a new Ks_NAF1 is required then a new Ks is required. But the same Ks may be used to derive a Ks_NAF2 for a different server NAF2. The variable in the key derivation is the NAF identifier NAF_Id_n. Of course, one could also think of deriving multiple Ks_NAF1 from one Ks for the same NAF1, but then you would need a new variable as input to the key derivation , e.g. additional state (like a counter) or protocol information (like nonces from the Ua protocol). I thought this would complicate things too much and left it out. The main scenario I wanted to address was to limit the load on the BSF and the HSS caused when the UE accesses a larger number of application servers in rapid succession, as explained in S3-040042.

To provide a flexible n value for each bootstrapping procedure, n value as the input parameter of key derivation should be delivered to UE after the successful bootstrapping procedure, so BSF can inform different n value associated to each NAF to the UE. Without a flexible method, we have to re-visit the decision made for S3-030743.

Guenther. I hope that my explanations above were able to address your concerns.